

FALL DETECTION FOR THE ELDERLY IN A SMART ROOM BY USING AN ENHANCED ONE CLASS SUPPORT VECTOR MACHINE

Miao Yu, Adel Rhuma, Syed Mohsen Naqvi and Jonathon Chambers

Advanced Signal Processing Group, Electronic and Electrical Engineering Department
Loughborough University, Loughborough, Leicester, UK
{elmy, a.rhuma, s.m.r.naqvi, eljac}@lboro.ac.uk

ABSTRACT

In this paper, we propose a novel and robust fall detection system by using a one class support vector machine based on video information. Video features, including the differences of centroid position and orientation of a voxel person over a time interval are extracted from multiple cameras. A one class support vector machine (OCSVM) is used to distinguish falls from other activities, such as walking, sitting, standing, bending or lying. Unlike the conventional OCSVM which only uses the target samples corresponding to falls for training, some non-fall samples are also used to train an enhanced OCSVM with a more accurate decision boundary. From real video sequences, the success of the method is confirmed, that is, by adding a certain number of negative samples, both high true positive detection rate and low false positive detection rate can be obtained.

Index Terms— voxel person, multiple cameras, one class support vector machine, fall detection

1. INTRODUCTION

There has been increasing public concern related to fall detection in recent years. According to the recent survey, falls are the leading cause of death due to injury among the elderly population and 87% of all fractures in this group are caused by falls. Although many falls do not result in injuries, 47% of non-injured fallers can not get up without assistance and this period of time spent immobile also affects their health. So, it is important to notice that detecting a fall event at home is an indispensable part of elderly people's care.

The most popular fall detection techniques involve:

a) Wearable or portable sensor-based method: For this type of method, a wearable sensor, such as a tri-axial accelerometer, is used. The acceleration information is collected and used for fall detection by different methods, such as threshold-based algorithms [2] [3] or some more complex pattern classification algorithm, such as a support vector machine (SVM) in [4]. Nowadays, it is a trend that people embed an accelerometer in a cell phone [5], which can detect a fall as well as reduce the inconvenience in requiring that an elderly person wears an extra accelerometer type sensor. The main disadvantages of this type of method include: 1. Old people sometimes forget to wear the sensors and some of the wearable sensors are not very convenient for old people to wear. 2. This type of method usually introduces a high false positive rate in fall classification.

b) Sound or vibration sensor-based method: In [6], M. Alwan et al. propose a fall detection system based on a passive floor-vibration sensor which records the signal generated from the floor's vibration when a person falls over, and some characteristics of the recorded signal, such as its amplitude, can be used for determining a fall event. In [7], M. Popescu et al. use an acoustic fall detection system, FADE, composed of a microphone array and a motion detector to determine falls by classifiers constructed from examples from one class (i.e.,

non-fall sounds). The problems for this category of fall detection methods include: 1. For different types of floor, the generated vibration signals will vary. 2. Some background noises will affect the sound signals obtained by the microphone arrays which will lead to ambiguity in fall determination.

c) Computer vision-based methods: There are various ways to detect a fall event using computer vision and signal processing techniques. In the papers [8] and [9], C. Rougier et al. use a threshold-based algorithm to compare the values of the extracted features with the corresponding thresholds to make decisions. The head's 3-D velocity and human shape information are extracted as features respectively. C. Juang and C. Chang [10] use an elegant self-constructing neural fuzzy inference network for posture recognition to detect a fall. As an effective tool for the classification problem, the SVM technique is applied in [11], the extracted features are finally fed to a multi-class SVM for precise classification of motions and determination of a fall event. In [12], a layered hidden Markov model (LHMM)-based approach is proposed to determine the state of the person (walking or falling) from a multiview pose classification strategy. For computer vision-based methods, the elderly person need not wear sensors and compared with the sound or vibration sensor-based methods, computer vision-based methods are not affected by the environmental noises that the sound or vibration sensors will suffer, so that the computer vision-based methods are in this sense superior compared with the other two categories. However, currently, the prevailing computer vision-based methods, such as the ones mentioned above, either construct different models for different activities [10] and [12], or build a very complex structure to distinguish falls from other activities such as the multi-class SVM method in [11]. Our work is underpinned by the observation that the fall activity shares similarities and can be ascribed to one class. This motivates us to use the one class classification technique for fall detection. In [13], M. Yu et al. propose the idea of using a one class classifier for fall detection and different one class classifiers are compared, the results show that the OCSVM achieves the best performance by obtaining the largest geometric means defined as $\sqrt{TPR * (1 - FPR)}$, where TPR is the true positive detection rate and FPR is the false positive detection rate, while being most robust to noise. In this paper, we show that by using an enhanced version of OCSVM incorporating certain negative (non-fall) examples, a more accurate decision boundary is obtained compared with the OCSVM using the target samples only. For the video features, a voxel person is constructed and the corresponding 3-D video features (changes of centroid position and orientation over one second) are extracted. These features then form the input to the OCSVM. Four cameras are used to better obtain the 3-D features and construct a more accurate voxel person as compared with using only two cameras. The structure of this paper is as follows: Section II describes how the video features are extracted. The OCSVM and enhanced OCSVM with negative examples are introduced in Sec-

tion III. Some experimental simulations are presented in Section IV. Conclusions and suggestions for future work are given in Section V.

2. VIDEO FEATURE EXTRACTION

We use 3-D features for the video features needed for the construction of the classifier. Traditional 2-D features, such as the bounding box aspect ratio [9] and projection histogram [10], have the following two major drawbacks: Firstly, they are sensitive to the distance of the person to the camera. Secondly, they are very sensitive to movements and falls in different directions. So, 2-D features are not efficient to be used for detection of falls that happen in different places and directions.

We use a multiple camera scheme to obtain the 3-D features [14]. Four color video cameras located at the corners of a room environment are used and we use the codebook subtraction technique [15] to obtain the foreground human body region. The advantages of the codebook background subtraction algorithm is that it is effective in compressed video frames and can deal with the problems of illumination change, and the change of background after training; besides, compared with other methods, such as mixture-of-Gaussians (MoGs) [16] and kernel based (KB) methods [17], its computational time is less as discussed in [15]. After background subtraction, some post-processing operations proposed in [18] are used to fill small holes, remove small blobs and smooth the boundary of the human body blob so as to improve the final results.

The voxel person is constructed from the background subtraction results of the cameras' frames. For voxel person construction, initially, we divide our 3-D room space into fixed size voxels ($2.5\text{cm} \times 2.5\text{cm} \times 2.5\text{cm}$), which are nonoverlapping cubes. Figure 1 shows the discretization of the 3-D space. The solid blocks represent the voxel blocks.

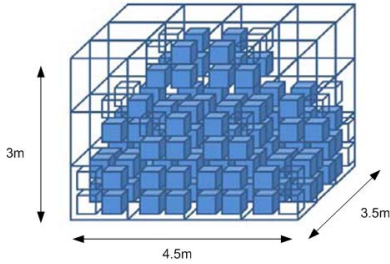


Fig. 1: The discretization of the 3-D room space with the dimension $4.5\text{m} \times 3.5\text{m} \times 3\text{m}$, the main part of this figure is cited from [14]

From the 2-D coordinate of a pixel in the recorded frame $[\ell_x, \ell_y]^T$, the undistorted coordinate $[u_x, u_y]^T$ on the focal plane can be obtained from the camera calibration [19]. Moreover, we can also obtain the rotation matrix R and translation vector \mathbf{t} , which reflect the relationship between the real world coordinates system and a camera-centered coordinate system. After obtaining these values, we can translate the origin of the camera-centered coordinate system $[0, 0, 0]^T$ and the point on the focal plane $[u_x, u_y, f]^T$ into the corresponding 3-D real-world coordinate by:

$$\mathbf{z} = \mathbf{R}^{-1}(\mathbf{c} - \mathbf{t}) \quad (1)$$

where \mathbf{z} is the real world coordinate and \mathbf{c} is the coordinate in the camera-centered coordinate system.

So, for a pixel, a ray can be constructed and we can identify a set of voxels that this ray intersects. The procedure is repeated for every

pixel in the 2-D frame for each camera and we can thereby obtain a pixel-voxel table for each camera.

The flow-chart of the 3-D voxel person reconstruction is shown in Figure 2.

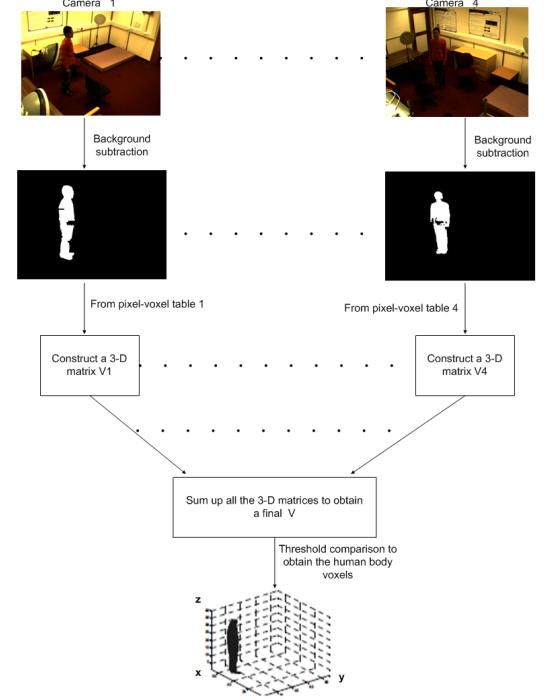


Fig. 2: The procedure of constructing the voxel person by using multiple cameras

When the silhouettes of a person in the images recorded by four cameras are extracted, for each image, we set the 3-D space voxels corresponding to the pixels in the silhouette with the value '1' according to the constructed pixel-voxel table, the remaining voxels are set to be '0'. In this way, a 3-D binary matrix \mathbf{V}_t^i is constructed for the i th camera at time t .

Four 3-D matrices are obtained for the four cameras and we sum up the four matrices to obtain a final matrix, \mathbf{V}_t , with $\mathbf{V}_t = \sum_{i=1}^4 \mathbf{V}_t^i$. A threshold is set to be th and if an element's value is no less than th , then the corresponding voxel is taken as a voxel in the 3-D human body region. From experimental evaluation a suitable threshold th has been found to be 2 and this is justified since two cameras are sufficient to recover the person within room space. This fusion of four cameras is an effective way to overcome the problem of occlusion or loss of the human body in some of the cameras. Because if the human body part is visible in at least th cameras we can still get a reconstructed voxel person correctly.

We next consider the extraction of the video features.

2.1. Video feature extraction

After we obtain the voxel person \mathbf{V}_t , as proposed in [14], we can extract two features which are useful for fall detection. They are the differences of the centroid position and a value called $gpsim_t$ reflecting the similarity of the voxel person's primary orientation over one second (usually a fall lasts 1-3s as highlighted in [20]).

The centroid of the voxel person at time t , $\mathbf{u}_t = [x_t, y_t, z_t]$ can be obtained by:

$$\mathbf{u}_t = \left(\frac{1}{M}\right) \sum_{j=1}^M \mathbf{V}_{t,j} \quad (2)$$

where M is the number of voxels belonging to the human body region.

The difference in the centroid position over one second is chosen as a feature for fall recognition. The horizontal difference of the centroid can be calculated as: $\sqrt{x_{t+1}^2 + y_{t+1}^2} - \sqrt{x_t^2 + y_t^2}$ and the vertical difference is: $|z_{t+1} - z_t|$. The two values are grouped into a video feature vector called ' $v_feature1'$ '.

The sample covariance matrix used to define the eigen information is:

$$\left(\frac{1}{M}\right) \sum_{j=1}^M (\mathbf{V}'_{t,j} - \mathbf{u}_t)(\mathbf{V}'_{t,j} - \mathbf{u}_t)^T \quad (3)$$

where $(\cdot)^T$ denotes vector transpose.

The eigenvalues and orthonormal eigenvectors of the covariance matrix are calculated and the eigenvector corresponding to the largest eigenvalue at time t is denoted as $eigenvect_t$ and a value denoted by $gpsim_t$ is calculated by:

$$gpsim_t = \max(eigenvect_t \cdot \langle 0, 0, 1 \rangle^T, -eigenvect_t \cdot \langle 0, 0, 1 \rangle^T) \quad (4)$$

where (\cdot) represents the dot product.

If the person is upright, the value is near unity; if he or she is on the ground, the value is near zero. The value of $gpsim_t$ is in the range of $[0, 1]$.

The difference of the value over a time interval (one second) is used as another video feature, which is denoted as ' $v_feature2'$ '. The resulting two features, ' $v_feature1'$ ' and ' $v_feature2'$ ' are used to form the inputs to the OCSVM.

3. ONE CLASS SUPPORT VECTOR MACHINE CLASSIFICATION

The one class support vector machine (OCSVM) is proposed in [21]. The basic idea behind OCSVM is that given a data set drawn from an underlying probability distribution P for the minority class, the OCSVM estimates a function f that is positive in a region S and negative in the complement, where S is the 'most-likely region'—a subset of the input space such that a test point drawn from P lies outside of S equals some a priori specified value between 0 and 1. In the application of fall detection, the minority class corresponds to falling and the majority class corresponds to various non-fall activities, we obtain the training samples for the falling class and use them to train an OCSVM classifier, the classifier can then capture the 'most-likely' region of the falling class. And if the test point is within this region, it is recognized as fall; otherwise, it is regarded as non-fall.

For a non-separable dataset, a kernel OCSVM [21] can be used to separate the mapped dataset in a high dimension feature space. The strategy of a kernel OCSVM is to map the training data into the feature space $\mathbf{x}_i \rightarrow \Phi(\mathbf{x}_i)$ to separate them from the origin with maximum margin.

The popularly used kernels are:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad (\text{Polynomial}) \quad (5)$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x} - \mathbf{y}\|^2} \quad (\text{Gaussian}) \quad (6)$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta) \quad (\text{Tangent}) \quad (7)$$

To design the classifier, we try to solve the following quadratic problem:

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{h}, \rho} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu\ell} \sum_i h_i - \rho \\ \text{subject to} \quad & (\mathbf{w} \cdot \Phi(\mathbf{x}_i)) \geq \rho - h_i, \quad h_i \geq 0 \end{aligned} \quad (8)$$

Here, $\nu \in (0, 1]$ and ℓ is the number of training data samples. The nonzero slack variables h_i are introduced to allow for the possibility of outliers (the data points which are not drawn from the distribution P).

For a new test point \mathbf{x} , the decision function is:

$$f(\mathbf{x}) = \text{sgn}((\mathbf{w} \cdot \Phi(\mathbf{x})) - \rho) \quad (9)$$

where $\text{sgn}(\cdot)$ is a sign function which yields the sign of the term in the bracket.

The optimization problem (8) can be converted to a dual problem as:

Using multipliers $a_i, b_i \geq 0$, we introduce an Lagrangian [22]:

$$\begin{aligned} L(\mathbf{w}, \mathbf{h}, \rho, \mathbf{a}, \mathbf{b}) = & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu\ell} \sum_i h_i - \rho \\ & - \sum_i a_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) - \rho + h_i) - \sum_i b_i h_i \end{aligned}$$

We set the derivatives with respect to the primal variables $\mathbf{w}, \mathbf{h}, \rho$ equal to zero respectively and obtain:

$$\mathbf{w} = \sum_i a_i \Phi(\mathbf{x}_i) \quad (10)$$

$$a_i = \frac{1}{\nu\ell} - b_i \leq \frac{1}{\nu\ell} \quad (11)$$

$$\sum_i a_i = 1 \quad (12)$$

According to the KarushKuhnTucker conditions (KKT) [22], the following constraints are satisfied:

$$a_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) - \rho + h_i) = 0 \quad (13)$$

$$b_i h_i = 0 \quad (14)$$

We substitute equations (10), (11) and (12) into the Lagrangian function and obtain:

$$\begin{aligned} L(\mathbf{w}, \mathbf{h}, \rho, \mathbf{a}, \mathbf{b}) = & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu\ell} \sum_i h_i - \rho \\ & - \sum_i a_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) - \rho + h_i) - \sum_i b_i h_i \\ = & \frac{1}{2} \sum_{ij} a_i a_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) + \frac{1}{\nu\ell} \sum_i h_i - \rho \\ & - \sum_{ij} a_i a_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) + \rho \\ & - \frac{1}{\nu\ell} \sum_i h_i + \sum_i b_i h_i - \sum_i h_i + \sum_i b_i h_i \\ = & -\frac{1}{2} \sum_{ij} a_i a_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \end{aligned}$$

So, a dual problem is obtained as:

$$\begin{aligned} \min_{\mathbf{a}} \quad & \frac{1}{2} \sum_{ij} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & 0 \leq a_i \leq \frac{1}{\nu\ell}, \quad \sum_i a_i = 1 \end{aligned} \quad (15)$$

Here a_i is the component of vector \mathbf{a} , $k(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$ and it is called the 'kernel function' defined as (5), (6) or (7).

According to the KKT condition [22], the decision function follows as:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_i a_i k(\mathbf{x}_i, \mathbf{x}) - \rho\right) \quad (16)$$

For the value of ρ , we can obtain it from the KKT condition. According to equations (13) and (14), we can see if a_i and b_i are non-zero, the corresponding pattern \mathbf{x}_i satisfies:

$$\rho = (\mathbf{w} \cdot \Phi(\mathbf{x}_i)) = \sum_j a_j k(\mathbf{x}_j, \mathbf{x}_i) \quad (17)$$

Here we propose another enhanced version of OCSVM with negative training samples (non-fall samples in our case), it has the following form:

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{h}, \rho} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu_1 \ell} \sum_i h_i + \frac{1}{\nu_2 \ell} \sum_l h_l - \rho \\ \text{subject to} \quad & \forall i \in P \quad (\mathbf{w} \cdot \Phi(\mathbf{x}_i)) \geq \rho - h_i \\ & \forall l \in N \quad (\mathbf{w} \cdot \Phi(\mathbf{x}_l)) < \rho + h_l \quad h_i, h_l \geq 0 \end{aligned} \quad (18)$$

Following a similar procedure, we can obtain a dual problem as:

$$\begin{aligned} \max_{\mathbf{a}} \quad & \sum_{i \in P, j \in N} a_i a_l k(\mathbf{x}_i, \mathbf{x}_l) - \frac{1}{2} \sum_{i \in P, p \in P} a_i a_p k(\mathbf{x}_i, \mathbf{x}_p) \\ & - \frac{1}{2} \sum_{j \in P, n \in N} a_i a_m k(\mathbf{x}_j, \mathbf{x}_n) \\ \text{subject to} \quad & 0 \leq a_i \leq \frac{1}{\nu_1 \ell}, 0 \leq a_l \leq \frac{1}{\nu_2 \ell} \\ & \sum_{i \in P} a_i - \sum_{l \in N} a_l = 1 \end{aligned} \quad (19)$$

where P is the index set for the positive samples (samples corresponding to the minority class) where N is the index set for the negative samples (samples which are not in the interested class).

The value ρ can be obtained as for an unbounded support vector \mathbf{x}_i with $0 < a_i \in P < \frac{1}{\nu_1 \ell}$ or $0 < a_i \in N < \frac{1}{\nu_2 \ell}$:

$$\rho = \sum_{j \in P} a_j k(\mathbf{x}_j, \mathbf{x}_i) - \sum_{l \in N} a_l k(\mathbf{x}_l, \mathbf{x}_i) \quad (20)$$

From the obtained \mathbf{a} and ρ , the decision function is obtained as (21):

$$f(\mathbf{x}) = \text{sgn}(\sum_{i \in P} a_i k(\mathbf{x}_i, \mathbf{x}) - \sum_{l \in N} a_l k(\mathbf{x}_l, \mathbf{x}) - \rho) \quad (21)$$

After the derivation of the dual problem, ρ and the decision function of the enhanced OCSVM. We next evaluate both the conventional and enhanced OCSVMs in the context of real fall detection.

4. EXPERIMENTS AND EVALUATIONS

The experiments were carried out in Loughborough University's Smart Room. There are four cameras located at the corners for which the union of their covering spaces is the whole room space. The four cameras are connected to four PCs, with one being a server and others being clients. The StreamPix 3 software [23] is installed on the PCs to perform video recordings and the format of the obtained video is AVI. Video recordings are converted to consecutive 320*240 frames for further processing by MatLab. A synchronizer which is connected to the server is used to ensure the synchronization of the cameras' recordings.

Figure 3 shows the layout of our smart room. And Figure 4 shows the hardware and software settings of our fall detection system.

A stuntman simulates the fall and non-fall activities, and three video sequences are recorded by each of the four cameras respectively. The first video sequence containing 30 fall activities and some non-fall activities is used for training with the corresponding video features being extracted manually from the sequences recorded by

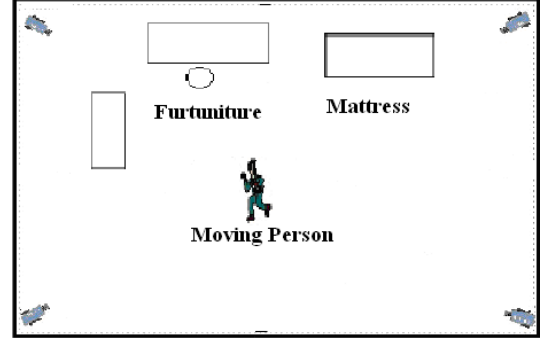


Fig. 3: The layout of the Intelligent Lab with the dimension of $4.5m \times 3.5m$

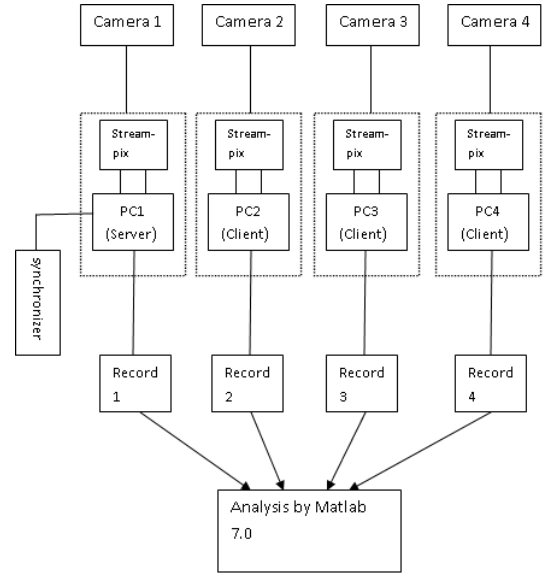


Fig. 4: The hardware and software settings for our fall detection system

the four cameras. The second video sequence containing 29 fall activities and 29 non-fall activities is for validation purpose—to tune the parameters of the OCSVM. Finally, a test video sequence which also contains 29 fall activities and 29 non-fall activities is recorded and used for evaluation of the performances of the constructed classifiers. All types of fall activities, like frontal falls, backward falls and side falls, are included. For the non-fall activities, the normal daily activities including walk, jog, crouch, sit and stretch are concerned.

Figure 5 shows the images recorded by four cameras and the corresponding background subtraction results. The human body is not seen in one of these cameras. The constructed voxel person is visualized in Figure 6.



Fig. 5: The recoded frames by four cameras and the corresponding background subtraction results

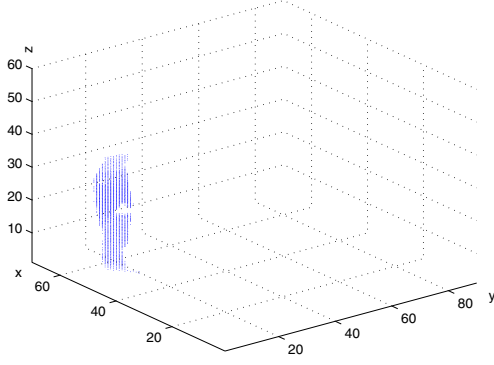


Fig. 6: The visualized 3-D voxel person construction result

For feature extraction, the voxels belonging to the human bod obtained from the background subtraction results by the steps sl in the flow chart in Section 2. Since a pixel-to-voxel table is pre-built for every camera, the computation amount for obtaining the human body voxels after background subtraction only involves looking up these four tables so that it can be real-time. Corresponding features can then be extracted from these human body voxels.

The obtained features are fed into our OCSVM for training, validation and testing. For a OCSVM, proper parameter values for ν_1 , ν_2 and γ should be set. To tune the parameters, an additional validation dataset is used and a grid search method is applied to get the optimal parameters which maximize the geometric means defined as $\sqrt{TPR * (1 - FPR)}$ for this validation dataset.

We give an example of tuning the parameters for an enhanced OCSVM with 7 negative samples, we set $\nu_1 \sim [0.005, 1]$, $\nu_2 \sim [0.05, 5]$ and $\gamma \sim [0.1, 10]$ and each parameter's range of its logarithm value is equally divided into 10 divisions for grid searching.

We find that the optimal parameter values corresponding to the largest geometric means value 0.9097 are $\nu_1 = 0.0416$, $\nu_2 = 0.05$ and $\gamma = 10$ after the grid search procedure. And after choosing the optimal parameters, we test the performance of this enhanced OCSVM under a test video sequence which contains 29 fall activities and 29 non-fall activities. The result is shown in Table 1.

Table 1: Performance of the enhanced OCSVM under the optimal parameter set

Activities	Fall	Non-fall
Detected as fall	100%	0%
Detected as non-fall	0%	100%

The true positive rate is 100% and the false positive rate is 0% thus the geometric mean can reach unity, which achieves the best performance.

Figure 7 shows the visualized feature classification results in 3-D space by using conventional OCSVM and enhanced OCSVM (using 7 negative samples) respectively. The first row of Figure 7 shows the training samples for the conventional OCSVM (left) and enhanced OCSVM (right) respectively (the blue circle represents the positive training feature samples corresponding to fall and the red star represents the negative training feature samples corresponding to non-fall and the x,y and z axis show the coordinates the 3-dimensional feature vector). Some testing feature points are shown in the two figures in the second row. The green regions in the two bottom figures represent the obtained region for the interested minority class-fall in our case, by using conventional and enhanced OCSVM respectively. We can see that by adding a few negative training samples, the obtained enhanced OCSVM can achieve a more approximate decision boundary.

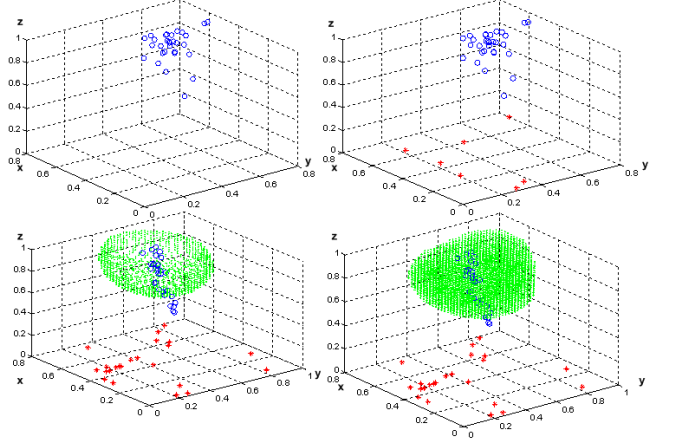


Fig. 7: The training samples and the corresponding decision regions for the conventional and enhanced OCSVM

We use different numbers of negative samples to train the OCSVM for comparison, Figure 8 shows the performance of the OCSVMs with different numbers of negative samples for training. Note, the parameters of different OCSVMs are tuned to the optimal ones by using grid search.

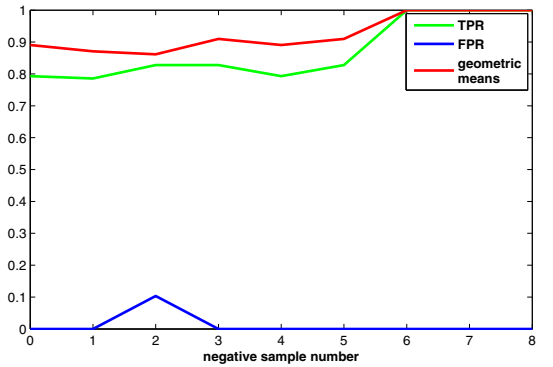


Fig. 8: The FPR, TPR and geometric means performances of OCSVMs trained with different numbers of negative samples

From Figure 8, we can see that for the conventional OCSVM the corresponding TPR is low. That is due to the fact that the training points are tightly clustered (as what is shown in the left-top sub-figure in Figure 8) so that the obtained OCSVM classifier overfits to the training points while the obtained decision region is small and the likelihood for a test point being in the decision region is low. If we use some negative training samples for training the enhanced OCSVM as proposed in the paper, as the number increases, the decision boundary will become more and more accurate, the TPR and FPR will be 100% and 0% respectively, and the optimal geometric means can be obtained.

Here we show another example in Figure 9 by using a training dataset with some outliers, due to the existence of these outliers, the obtained decision region is large. We can see by only adding one negative samples and training the enhanced OCSVM, the corresponding decision region will become tight and a more accurate result is obtained.

As such, we can see that by only adding a small number of negative samples for training the OCSVM using the scheme proposed in this paper, the overfitting problem can be avoided while better generalization performance can be obtained, both high true positive detection rate and low false positive detection rate are achieved thus a high geometric mean can be obtained.

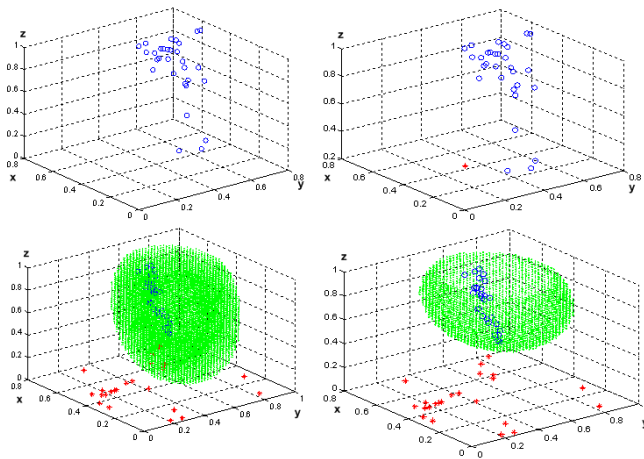


Fig. 9: The training samples and the corresponding decision regions for the conventional and enhanced OCSVM under a training dataset with outliers

5. CONCLUSION

In this paper, we have proposed a new fall detection method based on an enhanced OCSVM with 3-D features extracted from multiple cameras. A voxel person is extracted from the 2-D human silhouettes obtained from the images of different cameras. The differences of the centroid's position and the orientation angle over an interval of one second are used as 3-D features. An enhanced OCSVM with negative training examples is used as the classifier. The experimental results show as compared with the conventional OCSVM, by introducing some negative samples, the performance of the enhanced OCSVM can be improved in terms of a higher geometric means value, and a better generalization performance.

6. REFERENCES

- [1] "Old people- nmc facts and figures, UK," <http://www.medicalnewstoday.com/articles/142487.php>.
- [2] A. Bourke, C. Scanail, K. Culhane, J. O'Brien, and G. Lyons, "An optimum accelerometer configuration and simple algorithm for accurately detecting falls," *In Proceedings of the 24th IASTED international Conference on Biomedical Engineering*, pp. 150–154, 2006.
- [3] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jamsa, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait and Posture*, Vol. 28, pp. 285–291, 2008.
- [4] T. Zhang, J. Wang, P. Liu, and J. Hou, "Fall detection by wearable sensor and one-class SVM algorithm," *Lecture Notes in Control and Information Science*, Vol. 345, pp. 858–863, 2006.
- [5] T. Zhang, J. Wang, P. Liu, and J. Hou, "Fall detection by embedding an accelerometer in cellphone and using KFD algorithm," *International Journal of Computer Science and Network Security*, 2006.
- [6] M. Alwan, P. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," *The 2nd IEEE International Conference on Information and Communication Technologies: from Theory to Applications - ICTTA'06*, 2006.
- [7] M. Popescu and A. Mahnot, "Acoustic fall detection using one-class classifiers," *31st Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota, USA*, 2009.
- [8] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3d head tracking to detect falls of elderly people," *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pp. 6384–6387, 2006.
- [9] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," *In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pp. 875–880, 2007.
- [10] C. Juang and C. Chang, "Human body posture classification by a neural fuzzy network and home care system application," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 37, pp. 984–994, 2007.
- [11] H. Foroughi, A. Rezvanian, and A. Pazirae, "Robust fall detection using human shape and multi-class support vector machine," *Computer Vision, Graphics and Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on*, pp. 413–420, 2008.
- [12] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A lhm-based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, pp. 1522–1532, 2008.
- [13] M. Yu, S. Naqvi, and J. Chambers, "Video-based fall detection system by using the boundary method," *Submitted to IET image processing*.
- [14] D. Anderson, R. Luke, J. Keller, M. Skubic, M. Rantz, and M. Aud, "Modeling human activity from voxel person using fuzzy logic," *IEEE Transaction on Fuzzy Systems*, Vol. 17, pp. 39–49, 2009.
- [15] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, Vol. 11, pp. 172–185, June 2005.
- [16] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *Int. Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp. 246–252, 1999.
- [17] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *European Conf. Computer Vision*, Vol. 2, pp. 751–767, 2000.
- [18] R. Gonzalez and R. Woods, "Digital image processing, third edition," *Pearson Education, Inc.*, 2008.
- [19] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, Vol. 3, pp. 323–344, 1987.
- [20] X. Yu, "Approaches and principles of fall detection for elderly and patient," *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on*, pp. 42–47, 2008.
- [21] B. Scholkopf, J. Platt, J. Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, Vol. 13, pp. 1443–1471, 2001.
- [22] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California*, pp. 232–237, 1998.
- [23] "Norpix, digital video recording software," <http://norpix.com/products/streampix.php>.